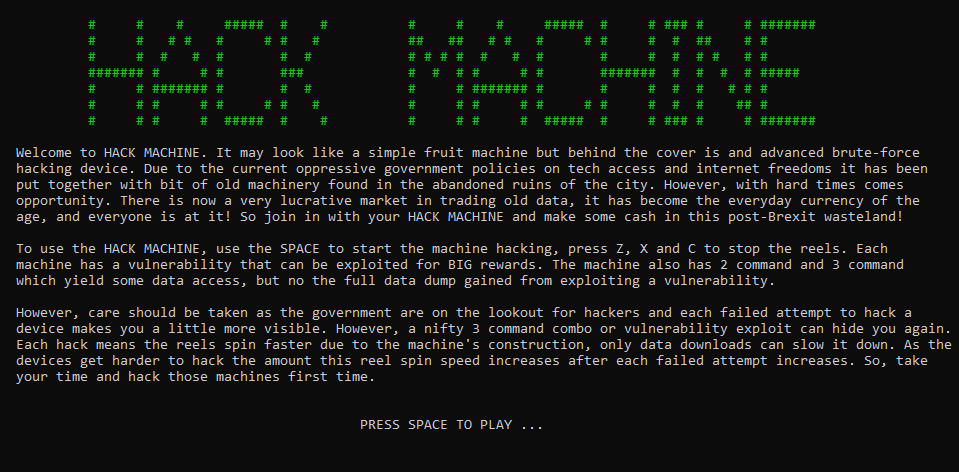
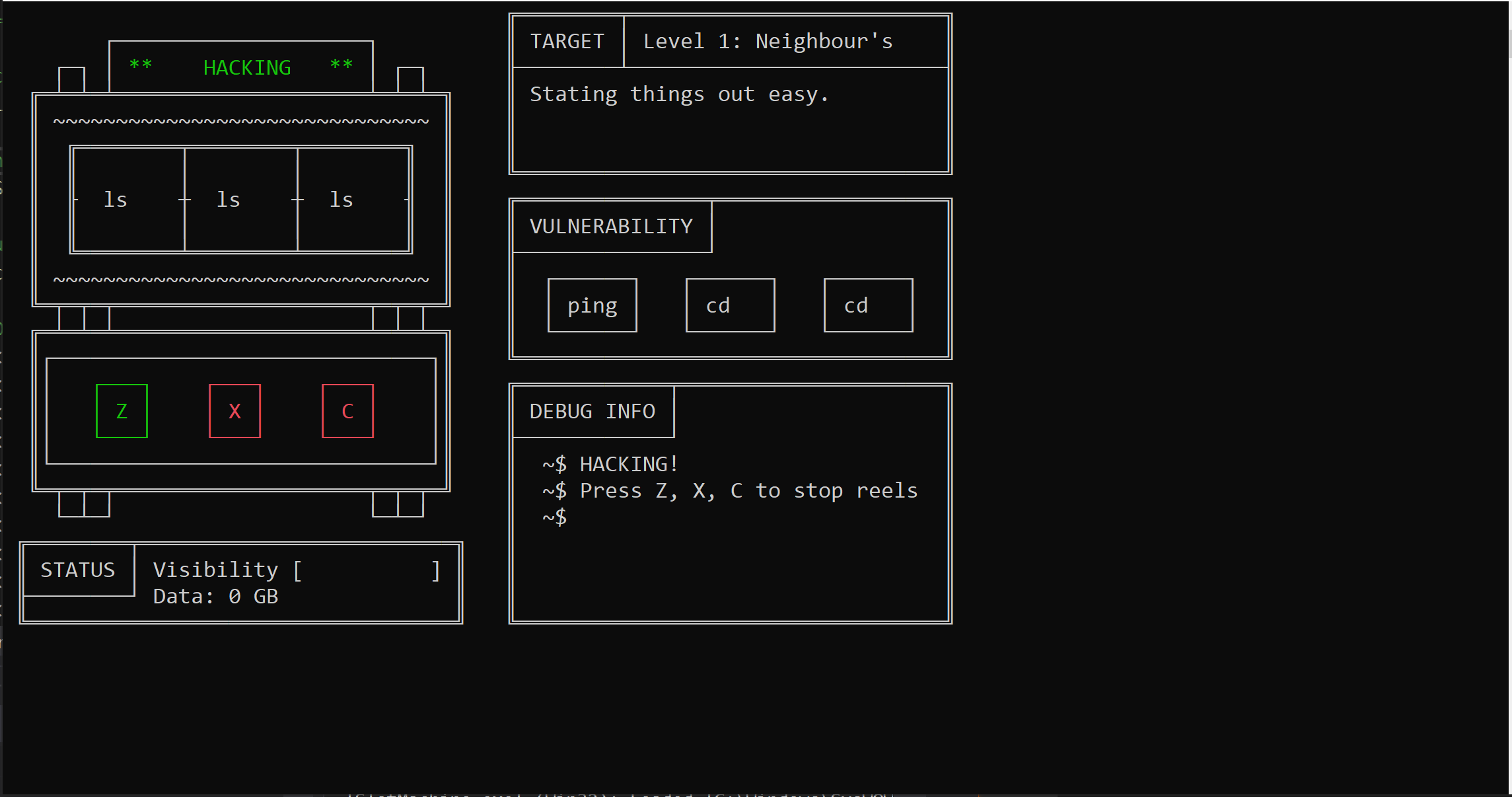
CMP104 –Fruit Machine Assessment Cover Document

Joseph Lee 1903399

## Game Overview

For this assessment rather then making a standard fruit machine, I decided to take the fruit machine concept and make a hacking themed fruit machine. So, the reel values are Linux commands and the prizes are data dumps. One feature I added was the randomly generated vulnerability victory condition that is randomly generated at the start of each spin. This vulnerability victory yields the highest prize but is also the hardest to achieve as it requires matching each reel to a specific value. In keeping with the theme this fruit machine game has been named ‘HACK MACHINE’, see screen shots of game below:





## Normal Game flow

A normal game would flow like below:

* Game intro and main screen displays with all the instruction games.
* **`Space`** to start game.
* Pints slot machine with animations.
* **`Esc`** to quit from this state to quit screen, `Space` starts reels spinning.
* All 3 reels keep spinning and vulnerability generated.
* Press **`Z`** key to stop Reel 1 spinning, other 2 reels keep spinning.
* Press **`X`** key to stop Reel 2 spinning, reel 3 keeps spinning.
* Press **`C`** key to stop Reel 3 Spinning.
* Once all reels stopped values are checked against victory conditions.
* Data value increased if a winning score, visibility increases.
* Each spin attempt reel spinning speed increases, this is more dramatic the higher the level.
* Winning will also reduce the spin speed increase.
* If Max visibility reached, game is over.
* If game is over **`R`** can be pressed to restart the game. **`Esc`** to quit from this state to quit screen

## Design Choices

The Below section outlines the main design choices made for this fruit machine.

Game Flow

The main game flow will be controlled by a state machine, where the game enters different states within the game loop, this allows the game flow to easily be changed simply by changing the game state variable. It is also very flexible, allowing each state to contain a loop to control things like animations etc. This structure also made it easy to build the game in sages and would allow easy expansion later if needed.

Theme

The fruit machine game theme has been chosen as a hacking theme, where each time the player plays the fruit machine, they are hacking a device. The Intro Screen sets out the theme and the basic controls. The target is displayed in the top right-hand column to add a sense of progress there are 6 levels. The players score is measured in data extracted from the target computers, witch a small prize for two matching reels, a big on for 3 matching reels and the biggest prize for matching the randomly Vulnerability generated at the start of the spin.

If a player fails to match any prizes on a spin, the machine’s spin speed will increase, and the player’s visibility will also increase. If this visibility reaches the top of the bar, then the player has been caught and the game is over.

In keeping with the theme, the in-game instructions are shown in a terminal like Debug Info window at the bottom-right of the game screen. This shows instructions on what to press next and if the player has won any prizes that spin.

Vulnerability

At the start of each spin, the game will randomly generate a vulnerability for the device being hacked. This will be a valid victory condition for that spin. This victory condition will yield the highest prize results, but due to the fact the three-reel values are often different and not matching, this victory condition can be very hard to achieve, especially at higher spin speeds. Due to its difficulty, in addition to the data prize, the spin speed is reset to its starting speed and the player visibility is reset to zero.

Target

To add a sense of progression, as the player’s data score increases the player progresses to higher levels. This makes the consequences for failing to match any reels more serious. The levels are shown in the below table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Level** | **Name** | **Description** | **Score (GB)** |
| 1 | Neighbour’s | Starting things out easy. | 0-9 |
| 2 | Small Office | Upping the takes? | 10-24 |
| 3 | Big Office | Now the real game begins! | 24-39 |
| 4 | Small Website | You’re no script kiddie now! | 40-54 |
| 5 | Large Website | You’re aiming big now | 55-69 |
| 6 | Government | AHHH! You’re crazy! | 70 or more |

## Game Flow

The game flow is controlled by transitioning the game between several states, the states are shown below:

1. **IDLE** - Plays slot machine idle animation.
2. **SPIN\_REELS\_123** - All 3 reels are continuously spinning until reel 1 key hit.
3. **SPIN\_REELS\_23** - Reels 2 & 3 continuously spinning until reel 2 key hit.
4. **SPIN\_REEL\_3** - Reel 3 continuously spinning until reel 3 key hit.
5. **ALL\_REELS\_STOPPED** - All reels have been stopped and victory state is checked. Game is then returned to IDLE.
6. **GAME\_OVER** - Max visibility has been reached; game is over. Displays Game over animations. Reset button is pressed to reset scores and return game to IDLE. Or Quit Button is pressed and game transitions to QUIT state.
7. **QUIT** - Game is quit, display quit message/ close console?

The below diagram shows an overview of how the game flows and transitions between these states. Further information on what happens at each state can be found in the README. For more information on this game’s code please consult the README file submitted with the game, this contains lots of detail on the concept and the implementation of each of the game’s features. The markdown (.md) and PDF versions have been included.

